

420KBG – Activité 01

L'objectif de cette activité est d'ajouter un service supplémentaire à une API existante, comprenant déjà plusieurs services. Cette API, nommée `API_SERVER`, est proposée par le sympathique Nicolas Chourot et vous la trouverez à l'adresse https://github.com/Nicolas-Chourot/API_SERVER

Idée générale

En détail, vous devez :

- Ajouter un contrôleur `BookmarksController` à l'API existante
- Ajouter une classe `Bookmark` modélisant un signet ou un favori (selon la nomenclature de votre fureteur Web de prédilection)
- Exploiter la classe `Repository` déjà dans le projet pour stocker les données à travers le contrôleur en s'inspirant de ce qui est fait avec `Contacts`, `Courses` et `Maths`

Une fois votre service écrit et testé, vous devrez l'héberger sur <https://glitch.com>

Classe `Bookmark`

Une instance de la classe `Bookmark` sera constituée des champs suivants :

Nom	« Type »
Id	Entier
Name	Chaîne de caractères
Url	Chaîne de caractères
Category	Chaîne de caractères

À propos des « types »

Nous pourrions faire des classes particulières pour certaines entités structurées (p.ex. : une URL), mais il est commun dans les systèmes à base de services de s'en tenir à une gamme de types relativement simples et universels, du moins dans les interfaces des services exposés.

Il y a plusieurs raisons pour cela . Par exemple, cela ouvre la porte à une plus vaste gamme de clients potentiels (sans égard au langage), et cela réduit le couplage de l'interface envers des règles sur lesquelles nous n'avons pas de contrôle. De manière surprenante, mêmes une entité aussi « simple » qu'un code postal est soumise à des règles « mouvantes ».

Dans un service exposé au monde extérieur, on vise habituellement à préserver les interfaces sur une longue durée, quitte à changer les implémentations en fonction des règles du moment.

Services à fournir

Votre API devra offrir les services suivants. Portez attention à la section Validation plus bas avant de les implémenter.

Verbe	URL	Rôle
GET	/api/bookmarks	Obtenir la liste des signets en ordre croissant d'Id
GET	/api/bookmarks?sort="name"	Obtenir la liste des signets en ordre croissant de nom
GET	/api/bookmarks?sort="category"	Obtenir la liste des signets en ordre croissant de catégorie
GET	/api/bookmarks/id	Obtenir le signet ayant le Id de valeur id
GET	/api/bookmark?name="nom"	Obtenir le signet ayant le Name de valeur nom
GET	/api/bookmarks?name="ab*"	Obtenir la liste des signets dont la valeur du champ Name commence par le préfixe ab (évidemment, ab n'est qu'un exemple, et tout préfixe non-vide serait valide)
GET	/api/bookmarks?category="sport"	Obtenir la liste des signets ayant un Category de valeur sport
GET	/api/bookmark?	Obtenir la liste des paramètres supportés pour une requête portant sur un seul signet (celles dont l'URL débute par /api/bookmark)
POST	/api/bookmarks	Ajouter un signet aux signets existants (note : vous devez valider le signet avant de l'ajouter)
PUT	/api/bookmarks/id	Modifier le signet ayant un Id de valeur id (note : vous devez valider la nouvelle valeur du signet avant de procéder à la modification)
DELETE	/api/bookmarks/id	Effacer le signet ayant un Id de valeur id

Validation

Certaines contraintes de validation devront guider votre travail. En particulier :

- Certaines *querystrings* peuvent contenir plusieurs paramètres. Par exemple, l'appel de service suivant devra lister les signets dont le nom débute par un 'e', triés en ordre croissant de nom, mais seulement ceux ayant "news" pour catégorie :

```
/api/bookmarks?sort="name" & name="e*" & category="news"
```

- Faites en sorte qu'il n'y ait pas de doublons d'Id
- Faites en sorte qu'il n'y ait pas de champs vides
- Faites en sorte qu'il n'y ait pas d'URL invalides

Modalités de remise

Organisation :	Travail individuel ou par équipe de deux
Date de remise :	Vendredi le 9 octobre 2020 à 23 h 59
Remise par Colnet?	<p>Oui, dans une archive zip nommée¹ comme suit :</p> <p>Nom-Prénom.zip (p. ex. : Tromblon-Gaetan.zip)</p> <p>...ou encore, dans le cas d'une équipe de deux</p> <p>Nom0-Prénom0--Nom1-Prénom1.zip</p> <p>Cette archive doit contenir :</p> <ul style="list-style-type: none">• Tous les fichiers sources• L'url de la version hébergée sur https://glitch.com/ dans le fichier html glitchLink.html

Amusez-vous bien!